Alarm Task Logic

Analysis

Mon, Jun 4, 2001

Initialization

Allocate and initialize global variables for `Alarms` Task. This includes setting `xACTIVP` pointers to `NULL`. Create message queue `ALRM` and pass to `NETCNCT`. This Acnet task name is used to receive alarm messages from nodes emitting D0 protocol alarm messages.

Normal loop

Await events 0 or 4. Set `OPTIONSA` byte to reading via pointer `ALOPTPTR`.

If event #0, perform usual alarm scan code.
- Copy time in cycle to `ELTIME`.
- If bit #6 of front panel option switches is clear,
  - Do `ALSCAN` to scan for all alarm conditions
  - Initz `ASCANNED`, `AINHIBIT`, `BSCANNED`, `BINHIBIT`, `CSANNED`, `NQUEUED`, `NETSFLG`.
  - If bit `00A1` set,
    - Do `ALRESET` to reset all alarms.
    - Do `ARSTMSG` to emit alarms reset comment message.
  - If bit `00A0` set,
    - Do `TRIPCLR` to clear all trip counts.
  - If `RESETDLY` = 0,
    - Copy beam status into sign bit of `BEAM` byte. (1=beam)
    - If `CACTIVP` = `NULL`,
      - Do `CSCAN`
      - If time for backup, set event #7 for later rebuild.
    - Else,
      - Allocate and initz active comment number list.
    - If `AACTIVP` = `NULL`,
      - Do `ASCAN`
      - If time for backup, set event #5 for later rebuild.
    - Else,
      - Allocate and initz active channel number list.
    - If `BACTIVP` = `NULL`,
      - Do `BSCAN`
      - If time for backup, set event #6 for later rebuild.
    - Else,
      - Allocate and initz active bit number list.
  - Else,
    - If `RESETDLY` had been = 1,
      - Set bit `00A1` to reset all alarms
      - Set `TRIPDSBL` = 3.
  - If bit `00A7` = 0,
    - If `NQUEUED` > 0,
      - Send event #2 to `Update` task to flush network queue
    - If `NETSFLG` != 0,
      - Do `NetSend` to flush to network
  - Else,
    - Clear `AINHIBIT`, `BINHIBIT`.
    - Check bit `00A7`, and if bit `00A7` is enabled, inhibit, bad
    - Increment `BINHIBIT`
- If `RESETDLY` = 0,

Do BEAMINH to inihibit beam if AINHIBT or BINHIBIT is nonzero.
If event #5, rebuild active analog channel list.

ASCAN flow

Registers:
D0.W= active channel loop counter
D1.W= #bytes/entry in ADATA table
D2.W= alarm flags
D3= scratch
D4= scratch
D5.W= counter for active entries
D6.W= current channel#
D7= n.u.
A0.L= ptr to current alarm flags field in ADATA entry
A1.L= ptr to current active channel# in active channels block
A2.L= ptr to ADATA table base
A3.L= ptr to FDATA table base
A4.L= scratch
A5.L= system globals
A6.L= Alarms Task variables ptr
A7.L= stackptr

ASNEW

On entry,
D6.W= current channel#
A0.L= ptr to current alarm flags field in ADATA entry
A6.L= Alarms Task variables ptr
On exit,
D2.W= current alarm flags
Registers D0-D6/A0-A2 preserved.